

Network-on-Chip: the intelligence is in the wire

Gerard Mas
STMicroelectronics
850, rue Jean Monnet
38926 Crolles, Cedex, France
Tel. +33-476-926-136
gerard.mas@st.com

Philippe Martin
Arteris
6 Parc Ariane – Immeuble Mercure
78284 Guyancourt, Cedex, France
Tel. +33-161-373-840
philippe.martin@arteris.net

ABSTRACT

In this paper we describe how Network-on-Chip (NoC) will be the next major challenge to implementing complex and function-rich applications in advanced manufacturing processes at 90nm technology and below. Although much progress has been made in connecting the various components encompassed in communication networks for System-on-Chip (SoC) designs (e.g. processor-specific bus architectures, interface adapters, etc.), until now there has not been the possibility to consider a chip-wide architectural methodology that takes into account all typical NoC design requirements.

As design complexity continues to increase, a more global approach is required to effectively transport and manage both on-chip and off-chip communication traffic, optimize wire efficiency and permit designs to scale in size, complexity, and IP block usage. The system described in this work is based on an innovative switching fabric that allows designers to build an efficient and scalable architecture for NoC implementation, without disrupting the existing IP interconnect solutions currently used in the design tools and flows.

1. INTRODUCTION

Two different forces mainly influence design methodology:

- A. The manufactory technology trends;
- B. The time-to-market.

1.1 Manufactory Technology Trends

Every year System-On-Chip designs become increasingly complex, while the associated number of transistors grows exponentially. Consequently, the functions integrated on the same chip (i.e., the IP blocks) have increased, and within the same silicon area it is now possible to implement designs addressing different applications, such as high-end computing, and mobile and Radio Frequency. Each of these electronic systems requires different technology options ranging from high-performance to low-power consumption, and featuring the characteristics of the target application. If nano-technology brings high integration and speed, one of the major drawbacks is that Non-Recurring Engineering (NRE) expenses have increased dramatically, inducing several adverse effects. The mask set manufacturing cost has been multiplied by a factor of ten in about three process technology generations. Therefore, the SoC design NRE costs range from 10M\$ up to 100M\$ for today's designs in 0.13 μ m technology, and they are bound to increase steadily as technology moves deeper into the nanometer regime. A radical change is needed to allow small-to-medium entrants in the market, or to

support products with volumes well below the multi-million chip threshold necessary to make a profit on low-cost ICs.

An issue common to all SoC designs is communication latency. In 65nm technology, it is predicted that the intra-chip propagation delay will be between six and ten clock cycles. Moreover, the increasing gap between processor clock cycle times and memory access times involves another form of latency hiding. Finally, co-processors can introduce additional latencies. Latency hiding is therefore a key aspect for achieving efficient parallel processing.

1.2 Hitting the On-Chip Traffic Wall

The challenge of on-chip communication traffic can be summarized in two ways:

1. Existing bus-based interconnect architectures and techniques are proving to be non-scalable, unable to meet leading edge complexity and performance requirements;
2. The physical medium of wires in 90nm and 65nm designs is becoming slower and not adequate for increasing operating frequencies.

As process geometries scale from 130nm to 90nm and below, the parasitic capacitance of wires becomes dominant with respect to gate capacitance and interconnect-induced delay dominates the gate delay. This issue is further aggravated because signals must be transported across longer distances and at higher speeds in more densely populated chips. The increased IP reuse and integration practice also create more "traffic" on narrower on-chip communication channels, not only necessitating to improve wire efficiency, but also of more effective design methods. This trend has forced designers to adopt different interconnection approaches such as heavily pipelined techniques. While this type of technique has proven effective in the past, as chip performance and complexity requirements increase, designers can no longer effectively manage the overall number of on-chip transactions. Innovative approaches are thus required to address this challenge.

1.3 Time-to-market

Some market segments in microelectronics such as Personal Computer (e.g. Graphics) or consumer electronics (e.g. Digital Video Display) do not leave room for late products. The first new product available on the market will achieve about all market shares leaving almost nothing to the followers. In such tight and competitive time-to-market context, in order to fill the design productivity gap, the industry is pushed to an extensively reuse practice, where everything, including existing silicon, is reused, thus introducing the new era of the System-in-Package. This new multi-chip integration of different dies connected within a single

package requires a standard and efficient way to link dies together, and also to anticipate future functionality implementation on a single die.

However, IC designers have found that traditional interconnect techniques employed in the previous technology nodes are no longer capable to cope with such increased on-chip traffic requirements. Minor evolutionary advances in on-chip interconnection have been developed from traditional bus-based architectures, including tiered or multi-layered techniques. While these methods enable minor improvements over earlier approaches and are suitable for the majority of traditional SoC designs, they are proving to be largely inadequate for today's leading-edge applications, and cannot effectively handle the complexity of next generation mainstream SoCs, which will require from dozens up to hundreds of IP blocks integrated on the same die, with operating frequencies in the Gigahertz range. In such design context, a single bus – or even multiple synchronous busses – is impractical due to large wire loads and resistances that introduce slower signal propagation. Managing the communication between multiple on-chip busses imposes additional design constraints, and results in reduced performance and increased silicon area.

The Network-on-Chip described in this paper will allow a fast implementation of complex designs in nanometer technologies, it will reduce the efforts necessary to build a new system to a function linear in the size of the new components rather than spending too much time on a difficult integration, and finally it will allow a complete redesign, or major application changes, either of reused IP blocks, or of the entire system.

2. NoC ARCHITECTURE

The advanced Network-on-Chip technology used at STMicroelectronics and developed by Arteris is based on system-level network techniques to solve on-chip traffic transport and management challenges. As shown in Figure 1, **synchronous bus** limitations dictate system segmentation and tiered or layered type of bus architectures.

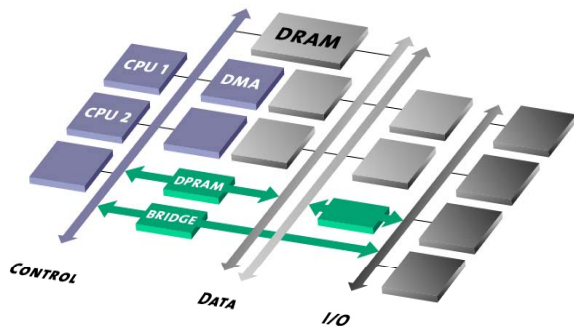


Figure 1. Traditional synchronous bus

In contrast, Arteris approach illustrated in Figure 2 is based on a homogeneous, **scalable switch fabric network**, which considers all the requirements of on-chip traffic.

The core of the NoC technology is the active switching fabric that manages multi-purpose data packets within complex, IP-laden

designs. The most important characteristics of this architecture can be summarized as:

- Packet switched approach [1][5][6];
- Flexible and user-defined topology;
- Globally Asynchronous Locally Synchronous (GALS) implementation.

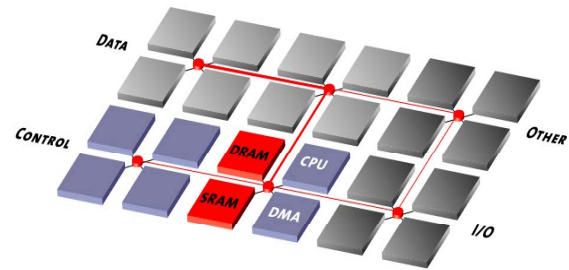


Figure 2. Switch fabric network

2.1 Network Architecture Strengths and Pitfalls

NoC strengths are inherently similar to those found in LANs and WANs, which have driven the evolution of standard busses such as PCI to move to “network-on-board” approaches like PCI-Express. Such advantages can be summarized as:

- A layered approach to communication [1], which isolates physical and transport implementation from higher level concerns such as transactions. By considering these issues independently, this method allows much higher clock speeds and higher throughput over fewer wires, as demonstrated in PCI-Express;
- Inherently scalable architectures with practically no limitation on the number of network agents, where the network topology can be devised using standard building blocks to solve each application specific problem such as found in [2] or [3];
- Once the network protocol is well defined and standardized, network components can be independently designed, fabricated and tested before being connected together. In SoC terms, it means that independent subsystems can potentially be designed separately from the specification, and implemented down to GDSII level before being assembled together, by simply using wires or abutment without creating any timing-related issues. This approach is possible because two devices connected to a network transport medium (for example an Ethernet cable) do not require external synchronization through a common clock, but can be synchronized through the transport medium itself.

However, care must be taken to avoid the pitfalls of too closely adopting the traditional approaches followed by long-distance communication networks [4], since the transport medium (i.e., optical fiber) is much more costly than the transmitter and receiver hardware, whereas in SoC the relative cost of wires and gates is different. Another factor impacting typical network trade-

offs is that long-distance networks are usually mostly focused on meeting bandwidth-related quality of service requirements, while several SoC applications are very sensitive to latency constraints.

As a consequence, a direct on-chip implementation of traditional network architectures would lead to significant area and latency overheads. For example the packet dropping and retry mechanisms that are part of TCP/IP require data storage, complex software control, and induce latency that would be prohibitive for most real-time, cost-sensitive SoCs. Therefore, different types of flow control mechanisms must be chosen for NoCs.

Another potential drawback of using network architectures to handle on-chip communication is to end up with a protocol that does not handle efficiently, if not at all, some features necessary for compatibility with existing IPs, which have to be integrated in the SoC design. Many networks do not have implemented in hardware the concept of a “read” transaction and can only perform “writes”, whereas several IPs, for example processors, do need a very efficient implementation of read transactions. The communication feature set of existing IPs, which a network architecture must be compatible with, is in fact quite large and includes burst types, atomic transactions, etc.

Therefore, a network architecture suitable for SoC designs must focus on leveraging the inherent strengths of network architectures, while avoiding the common overheads and the compatibility problems. This can only be achieved by carefully devising the various protocol layers involved (physical, transport, packet, transaction), and by considering the same constraints during the implementation of the network components.

2.2 Physical Layer Study

As already discussed above, a layered approach to communication enables a separate optimization of each layer, typically geared towards distinct end-user system benefits: for example physical layer usually provides raw bandwidth, while Quality-of-Service (QoS) features are mostly handled by the upper layers.

New low-level layers can be devised, allowing a close tracking of technological advances, while preserving the compatibility at application level. This approach has been exploited in the Ethernet physical layer, which has successfully transitioned from an initial 1MHz bus-oriented single coaxial cable, up to 10GB/s point-to-point, and now wireless implementation. At the same time, new upper layer levels may be developed to take advantage of existing infrastructure, for example ATM over SONET hardware.

In NoCs, the focus is to optimize the physical layer towards raw speed, low cost, and reliability. In fact, depending on the considered physical link requirements (required bandwidth, quality of service, physical distance span), the optimal physical link implementations will differ, just as they differ from short and cheap RJ45 wires to thousand-of-miles length of optical fibers across the hierarchy of LANs, enterprise LANs, and WANs.

Within NoCs scaling and cost factors between long-haul links and short links are not as radical as in a WANs, and common requirements can be established between the physical transports:

- *Optimized framing and flow control signals.*

Since these signals must be looked up typically at every cycle in order to perform any operation, their definition and timing must be carefully planned to allow high-speed

operation, including both their use by NoC components (i.e., switch arbiters) and their generation (i.e., flow control propagation). They must also provide an easy insertion scheme of pipeline stages when the physical span of the link makes the wire propagation time exceed the length of a single clock cycle.

- *High wire and gate efficiency.*

Dedicating wires to transport specific information costs both in terms of wiring area and associated logic or storage elements (buffering or pipelining). Packetizing and multiplexing this information on the same wires within the clock rate capability of the technology reduces hardware cost for a given system performance. At the same time, reducing the number of wires actually increases the maximum operation speed by reducing the load on the framing and control signals.

On the other hand, actual implementation will locally differ based on the link requirements:

1. The amount of multiplexing will depend on the required aggregate bandwidth and achievable clock rate;
2. The clocking scheme can be fully synchronous for short links, where timing closure between two synchronous endpoints is easily guaranteed, but may be asynchronous if both ends have different clocks, or if their alignment cannot be assured during the back-end phase;
3. The internal link flow control mechanisms used may vary from traditional handshaking to more complex schemes such as credit-based, depending on the QoS requirements and clocking scheme of the link.

In addition, specific features can be attached to chosen links: parity or CRC error detection at the receiving end can be implemented for further processing at the upper protocol layers. Such capabilities can be realized within high fault-tolerance systems, where IC transient errors become more and more important issues.

2.3 Other Architecture Characteristics

Similar studies and trade-offs have been performed on packet and transaction layers to derive a NoC architecture that is suitable for next-generation SoC designs. The principal characteristics of this NoC architecture are: 1) a packet switched approach; 2) a flexible and user-defined topology, which is also fully parametrizable in order to cover different SoC applications; 3) a GALS implementation. The most important benefits are the following:

Efficiency: a simple communication mechanism of packetized data requires less control logic, transport wires, and intermediate storage than interconnections based on complex pipelined protocols, thus resulting in higher operation speeds and reduced overall area.

Scalability: no permanent resource is allocated within the switch fabric for the entire duration of a given transaction; hence, it can handle as many IPs with as many concurrently on-going transactions as necessary for a given system performance target.

Flexibility: detailed packet format, network topology, and individual link throughput, can be user-defined based on application-specific characteristics. The NoC can thus match

application constraints, for example floorplanning or explicit dataflow quality of service requirements.

Compatibility: transaction-level layer implements a Load/Store model and has been carefully devised to be compatible with common busses or IP socket protocols (OCP [7], AMBA™ AHB/AXI [8] or custom-made property IP interfaces). Bridges to these protocols are available and can be freely mixed within a single NoC without changing the interconnect structure every time a new CPU core or IP library is used in a SoC design.

Timing convergence and routability: exclusive use of point-to-point links, combined with pipelining and simple transport flow control, ensure easy timing convergence even at high operation speeds. When the NoC spans a large area, a Globally Asynchronous Locally Synchronous approach can be used to guarantee timing convergence by reducing the area of separate clock domains, where locally synchronous on-chip sub-systems are globally connected asynchronously. This approach is becoming more and more necessary as the distributed RC delays of the global interconnects continue to dominate in new process generations, and it offers a much more comprehensive and effective way to optimize the communication network of an entire SoC design, thus targeting performances of 1GHz at 90nm.

The ability to reach timing closure at a global level is especially important to SoC designers, who currently struggle with fully synchronous approaches and several clock distribution limitations in complex VLSI ICs.

Point-to-point links and reduction in the number of wires also increase routability.

Application debug or monitoring: error management and on-chip packet tracing can be implemented in the design, enabling debug during software integration or software monitoring of application traffic statistics.

The NoC consists of a set of configurable module generators (switches, links, bridges, memory controllers, etc.) within a design environment that enables a fast network configuration. Moreover, the NoC design environment supports instantiation and parameterization of the modules, connections between the modules, protocol and performance analysis, early area and timing estimates, RTL (Register-Transfer-Level) and back-end view generation.

3. STATUS AND CONCLUSIONS

In this paper we presented a new Network-on-Chip technology, which is scalable and tailors any process technology, chip size, number of agents, and SoC architecture or topology. The NoC has been implemented on silicon in 0.13 μ m technology, and tested at frequencies up to 500MHz, using a standard CAD flow methodology.

This NoC technology is providing very competitive wiring, gate and power efficiency compared to achieved throughputs, with limited latency overhead. The globally asynchronous approach offers easy floorplanning and timing closure, and allows higher clock rates. Finally, embedded trace capabilities provide efficient driver or application software debug during the software integration phase.

A lot of intelligence can be put into the SoC wires by reconciling networking techniques with the cost and system performance constraints of SoCs.

4. REFERENCES

- [1] L. Benini and G. De-Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer*, vol. 35, pp. 70-78, Jan. 2002.
- [2] F. Gilbert, M. J. Thul, and N. When, "Communication Centric Architectures for Turbo-Decoding on Embedded Multiprocessors," in *Proc. DATE*, Mar. 2003, pp. 356-361.
- [3] J. Xu, W. Wolf, J. Henkel, S. Chakradhar, and T. Lv, "A Case Study in Networks-on-Chip Design for Embedded Video," in *Proc. DATE*, Feb. 2004, pp. 770-775.
- [4] A. Radulescu and K. Goossens, "Communication Services for Networks on Chip," in *Proc. SAMOS*, Jul. 2002, pp. 275-299.
- [5] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," in *Proc. DAC*, Jun. 2001, pp. 684-689.
- [6] P. Guerrier and A. Greiner, "A Generic Architecture for On-Chip Packet-Switched Interconnections," in *Proc. DATE*, Mar. 2000, pp. 250-256.
- [7] OCP International Partnership. Open Core Protocol Specification.
- [8] ARM. AMBA AXI Protocol Specification, Jun. 2003.